Neural Action Policy Safety Verification: Applicablity Filtering Technical Appendix

Primary Keywords: *None*

A Neural Action Policy

A (ReLU) feed-forward *neural network* over $\mathcal{S}$ is a (real-valued) function

$$f_\pi \colon \mathcal{S} \to \mathbb{R}^{d_d}, s \mapsto f_d(\dots f_2(f_1(s))),$$

where d denotes the number of layers in the NN, d_i for $i \in \{1, \dots, d\}$ denotes the size of layer i, and

- $f_1 \colon \mathcal{S} \to \mathbb{R}^{d_1}, s \mapsto (s(v^1), \dots, s(v^{d_1}))$ is the *input layer* function, where $v^j \in \mathcal{V}$ for $j \in \{1, \dots, d_1\}$ denotes the state variable associated with input neuron j.
- $f_i \colon \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}, V \mapsto ReLU(W_i \cdot V + B_i)$, for $i \in \{2, \dots, d-1\}$, is the function of *hidden layer* i. W_i is the weight matrix of layer i, i.e., $(W_i)_{j,k}$ denotes the weight of the output of neuron k in layer $i-1$ to the input of neuron j in layer i. B_i is the bias vector, i.e., $(B_i)_j$ denotes the bias of neuron j in layer i.
- $f_d \colon \mathbb{R}^{d_{d-1}} \to \mathbb{R}^{d_d}, V \mapsto W_d \cdot V + B_d$ is the function of output layer d. Here, no ReLU activation is applied.

Given a neural network f_π, a *neural action policy* is a function

$$f_{\mathcal{L}} \colon \mathcal{S} \to \mathcal{L}, s \mapsto \underset{\{l \in \mathcal{L} \mid \exists o \in \mathcal{O}_l \colon s \models o\}}{\operatorname{argmax}} f_\pi^l(s)$$

where f_π^l denotes the output of f_π associated with l (abbreviated π_l in the main text).

B Abstract Transition Problem in SMT

In this section, we outline the SMT encoding of the abstract transition problem, i.e., given operator $o = (l, g, u)$[1] does there exist a concrete state $s \in [s_{\mathcal{P}}]$ such that $s \models o$, $s[\![o]\!] \in [s'_{\mathcal{P}}]$ and $\pi(s) = l$. Importantly, our encoding differs from the encoding used by VEA only in the label selection of the policy.

Each state variable $v \in \mathcal{V}$, occurs in an *unprimed* form; representing the state variable in the source state and a *primed* form v' representing the updated state variable in the successor state.

[1]VEA apply SMT checks on a per operator basis and iterate operators as part of their search algorithm (Vinzent, Steinmetz, and Hoffmann 2022).

To encode the neural network structure we introduce *real-valued* auxiliaries variables:

$$\{v_{i,j} \mid i \in \{1, \dots, d\}, j \in \{1, \dots, d_i\}\}$$

and

$$\{v^{i,j} \mid i \in \{2, \dots, d-1\}, j \in \{1, \dots, d_i\}\}$$

corresponding to neuron inputs and outputs. More precisely, $v_{i,j}$ corresponds to the neuron output and $v^{i,j}$ to the input of hidden layer neurons. For $i = 1$, $v_{i,j}$ is syntactic sugar for the respective state variable v^j in the input layer.

The abstract transition problem is then encoded by the conjunction of the constraints:

(i) $lo_v \leq v$ and $v \leq up_v$ as well as $lo_v \leq v'$ and $v' \leq up_v$ for each $v \in \mathcal{V}$, where lo_v denotes the lower bound and up_v denotes the upper bound of state variable v.

(ii) p if $s_{\mathcal{P}}(p) = 1$ and $\neg p$ if $s_{\mathcal{P}}(p) = 0$ as well as p' if $s'_{\mathcal{P}}(p) = 1$ and $\neg p'$ if $s'_{\mathcal{P}}(p) = 0$ for each p in $\mathcal{P}$ where p' denotes the predicate in its primed form, i.e., with primed variables.

(iii) $\bigwedge_{i \in \{1, \dots, m\}} g_o^i$

(iv) $v' = u(v)$ for each $v \in \mathcal{V}$

(v) $v^{i,j} = \sum_{k=1}^{d_{i-1}} (W_i)_{j,k} \cdot v_{i-1,k} + (B_i)_j$ and $v_{i,j} = ReLU(v^{i,j})$ for each hidden layer $i \in \{2, \dots, d-1\}$ and each neuron $j \in \{1, \dots, d_i\}$,

(vi) $v_{d,j} = \sum_{k=1}^{d_{d-1}} (W_d)_{j,k} \cdot v_{d-1,k} + (B_d)_j$ for the output layer d and each neuron $j \in \{1, \dots, d_d\}$,

(vii) $\bigwedge_{l' \in \mathcal{L} \setminus \{l\}} \left(v_{d,j} > v_{d,k} \vee \neg \bigvee_{o \in \mathcal{O}_{l'}} \bigwedge_{i \in \{1, \dots, m\}} g_o^i \right)$ where $j \in \{1, \dots, d_d\}$ is the output neuron associated with l and $k \in \{1, \dots, d_d\} \setminus \{j\}$ is the output neuron associated with l' (abbreviated π_l and $\pi_{l'}$ in the main text).

(i) constrains the variables to respect the corresponding state variable domains, such that every satisfying assignment to the SMT encoding corresponds to a valid state pair s, s'. (ii) then encodes $s \in [s_{\mathcal{P}}]$ and $s' \in [s'_{\mathcal{P}}]$. (iii) encodes $s \models o$, and (iv) encodes $s' = s[\![o]\!]$. $\pi(s) = l$ is encoded by (v – vi, neural network) and (vii, label selection) – applicability of label l itself is entailed by $s \models o$ (iii).

Note that the presented encoding is specific to the NN-tailored solver *Marabou* (Katz et al. 2019) in that it assumes a special construct for ReLU constraints. Furthermore, *Marabou* only supports real-valued variables, i.e., integer state variables are continuously-relaxed. VEA establish integer support via a branch & bound loop around *Marabou* (Vinzent, Steinmetz, and Hoffmann 2022).

References

Katz, G.; Huang, D. A.; Ibeling, D.; Julian, K.; Lazarus, C.; Lim, R.; Shah, P.; Thakoor, S.; Wu, H.; Zeljic, A.; Dill, D. L.; Kochenderfer, M.; and Barrett, C. 2019. The Marabou Framework for Verification and Analysis of Deep Neural Networks. In *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *LNCS*, 443–452. Springer.

Vinzent, M.; Steinmetz, M.; and Hoffmann, J. 2022. Neural Network Action Policy Verification via Predicate Abstraction. In *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling, ICAPS 2022, Singapore (virtual), June 13-24 2022*. AAAI Press.